目 次

・・・・・0・・・・・

はじめに

部長 平野湧一郎

このたび NPCA に来て下さり、この部誌を取ってくれた皆さん、本当にありがとうございます。私は NPCA の現部長 平野湧一郎です。この一年間パソコンを買い換えたりとあわただしかったですが、なんとかここまでくることができました。

この部のことを紹介いたしますと、旧校舎の四階でパソコンでプログラミングや音楽や CG の作成、サーバーの管理などの活動をしております。

部室にくる決まった時間などはなく、朝学校があいてから夜の 6:00 に学校がしまるまでのたいていいつでも活動しています。現在稼動しているパソコンはサーバーを合わせて 6 台となっています。

あと、ホームページを運営しています。アドレスは `http://npca.my-sv.net/` になっています。

・・・・1・・・・

戸外でもノベルゲームで遊ぶ試み

61 回生 謎の魔人 X

前書き

こんにちは、謎の魔人 X です。X は半角です。この文章は何について書きたいかというと、Windows で動作するノベルゲームを Windows 以外のプラットフォームで動かしてみようよ、という取り組みです。Windows 以外のプラットフォームとは、携帯ゲーム機等も入っており、屋外でパソコン用のノベルゲームを楽しんだり、なんてことが出来ちゃいます。まぁ、そのために費やす労力を考えればあまり費用対効果は良いとはいえないかもしれないですが。それについて語っていきたいと思います。

どうでもいいですが、この文。締め切りかなりあやうい所で書いてたりします。前回もかなりあやうかったですが、今回はそれをさらに上回る状態です。なんせ、締め切りを迎えた時点で一文字たりとも書いていなかったもんですから。締め切り翌日なのにいきなりゼロから書くのは相当につらいです。まぁそういう訳なので、多少おかしな所があっても気にしないでください。急いでただけですので。

本題

さてそれでは本題。基本的に、ノベルゲームというのは、Windows で動くものが多いんですよね。先程からノベルゲームと言っていますが、これはつまり、その、なんと言うか、ギャルゲーとか、成人向けゲームとかが多かったりします (汗　そういう水商売ですので、マイナーなプラットフォームは常に相手にされません。市販のノベルゲームは殆ど Windows でしか動かないんですよね。それを Macintosh 等の他のプラットフォームで動かす、ということですが、普通の Windows のプログラムの場合、Windows の API を使ったり、DirectX を使ったりしていますから、他のプラットフォームに移植するのはかなり骨です。また、そのプログラムのソースを知らないと出来ないことも多々あるため、実質移植は困難...ということになるのですが。ノベルゲームに関しては運が良ければ他のプラットフォームでも遊べます。どういうことか説明しましょう。

ノベルゲームは基本的に背景 (立ち絵) があって、その上にメッセージがあってメッセージを読み進めていく、というのが基本のスタイルになります。ノベルゲームを作成するにあたって、そのようなプログラムを一から作っていってもいいのですが、ノベルゲームはメッセージを読み進めていくスタイルが似ているのでノベルゲームを作る人それぞれが似た部分を作る羽目になり、効率が悪いです。そこで、「ノベルゲーム用のプログラミング言語」というものを利用します。これをゲームエンジン (スクリプトエンジン) といいます。ノベルゲームに限らず、ゲームエンジンは RPG ツクール等、他のプログラム開発でも利用されています。このゲームエンジンを使えば、一からプログラムを作らなくても、文字の開始位置や行間・立ち絵等を決められたプログラミング言語

で記述することで、楽にノベルゲームを作成することが可能です。C 言語や DirectX などを知らなくても、ノベルゲームが作れる、というメリットがあります。そのゲームエンジンの対応しているプラットフォームが、実際のノベルゲームが遊べるプラットフォームとなるわけです。

とは言うものの、実際は Windows にしか対応していないものが多かったりします。そこで、他のプラットフォーム用の実行環境を用意してやれば、他のプラットフォームでも動きます。プログラミング言語の段階ではプログラムはテキストに記述された文字にすぎないので、それを解釈するようなプログラムを作ってやれば良い訳です。これにも割と手間はかかりますが、そのゲームエンジンを使用しているゲームが一気に遊べるようになります。それでは代表的なゲームエンジンをあげてみましょう。

NScripter	`http://www.NScripter.com/`
吉里吉里 2	`http://kikyou.info/`
AVG32	
RealLive(AVG2000)	
Ethornell	

などなどです。

このうち、NScripter、吉里吉里 (きりきり) は仕様が公開されており、例えば一般ユーザーがノベルゲームを作りたいなと思ったときにすぐに利用することが出来ます。下の３つは企業が使用しているゲームエンジンで、仕様は公開されていません。仕様が公開されているゲームエンジンに関しては、他のプラットフォーム用の実行環境を用意するのは比較的簡単ですが、仕様が公開されていないものに関してはそもそもそのプログラミング言語がどのような規則が記述されているかが分からないので、そこを予測しながら開発していくことになり、他のプラットフォーム用の実行環境を用意するのは比較的難しくなります。このうち、NScripter、AVG32、RealLive に関しては有志により実行環境が非公式で C 言語により開発されていますので、C 言語を使用出来るプラットフォーム（ほぼ全てです）なら、このゲームエンジンを利用したノベルゲームが遊べることになります。うまくやれば、PSP や iPod 等で動かすことも可能で、自分は iPod touch での動作を確認してます。如何せん画面が小さいですが、慣れるとさほど気にならないもんですね。まぁ、携帯の画面でもゲームができるくらいなんですから、当たり前かもしれませんが。吉里吉里、Etornell に関しては、実行環境がないので、他のプラットフォームで遊ぶのは無理です。そうでなくても、ここに挙げた５つのゲームエンジン以外の企業独自のゲームエンジンを使っているとかいうことも多々ありますし、正直言って遊べるゲームは少ないです。それでも、Macintosh や携帯ゲーム機で楽しめるノベルゲームがあるだけでもマシですけどね。どうしてもというなら、自分で実行環境を頑張って開発するという手もなくはないですが、かなりの労力がかかることが予想されます。ほかの手段として、別なゲームエンジンで動くノベルゲームを NScripter に移植して、NScripter の用の実行環境で動作させる、という手もあり、ゲームに応じたコンバータがいくつか公開されています。ちなみに、吉里吉里 2 の次のバージョンである吉里吉里 3 では、クロスプラットフォームを目指している、とのことです。そうなればいいなーとは思いますけど、かなり先の話になりそうですね。

それでは実際に代表的なスクリプトエンジンである NScripter についてみていきましょう。NScripter を使用しているノベルゲームで有名なのはひぐらしのなく頃にシリー

ズですね。他にも採用されているゲームはありますが、成人向けのものが多いのでここでは伏せておくことにします。知りたければ、Wikipedia に纏められているのがあるのでそれを。NScripter を他のプラットフォームで動かす実行環境として、ONScripter というものが開発されています。ONScripter は以下のサイトからダウンロードできます。

`http://ogapee.at.infoseek.co.jp/ONScripter.html`

ですが、オープンソースの世界では、ソースコードだけを配布してバイナリは配布せず、各自で勝手にビルド (コンパイル) していってください、的なノリのものが多いです。数種類のプラットフォーム向けのコンパイル済みのバイナリも数箇所で配布されているようなので、慣れないうちはそれを使いましょう。

ONScripter のソースをビルドするなり、バイナリを入手するなりしてきたら、ゲームのディスクから必要なファイルを取り出します。NScripter を使用して作成されたノベルゲームも、製品の CD を入れると普通にインストーラが立ち上がって遊べるようになっているので、NScripter であることを意識することは少なく、ここでインストーラは、ディスク内のファイルをリネームしたりしてファイルを正しい位置に単純にコピーする、ということをやっています。普通に遊ぶときはインストーラに任せればよいのですが、Windows 以外の環境を使用していたりする場合は手動でファイルを配置することになります。大抵の NScripter を使用したノベルゲームでは手動でのインストールができますが、ごくまれに出来ないこともありますので、その際は Windows 環境が必要になります。Windows を使っているのであれば、インストーラを使ってインストール先のディレクトリをそのまま持ってくることもできます。

NScripter の基本的なファイル構成は、スクリプト本体である `nscript.dat`(又は `0.txt`)、アーカイブファイルである `arc.sar`(又は `arc.nsa`)、そのほか BGM や効果音のファイルから成り立っています。

一度起動するとセーブデータ等のファイルも作成されるようになります。このファイル群をひとつのディレクトリにまとめ、そのディレクトリを ONScripter で指定してやることでゲームが起動します。ちなみに、「ひぐらしのなく頃に」ではファイルのコピー先が `filelist0.txt` に記述されていて、その通りに配置することになります。

`nscript.dat`(`0.txt`) は、スクリプト本体で、ここに記述されている通りにゲームは動作します。ということは、このファイルを見ればすべてのテキストを閲覧したりどういう仕組みになっているかを知ることができます。`0.txt` の場合は、記述されているコードそのまんまなのですが、`nscript.dat` の場合は暗号化されています。もとのコードは見られたくないということで、配布時は普通暗号化されていますが、簡単なプログラムで復号化して元のテキストデータに戻すことができます。復号化するには、NSDEC というソフトを使います。Windows 用のバイナリしか配布されていませんでしたが、大して難しいプログラムでもないので自分で作ればほぼどの環境でも復号化は可能です。ちなみに、ONScripter は自動で復号化を行ってくれるので単に遊ぶだけなら復号化の必要はありません。携帯ゲーム機で遊ぶ際は、文字が小さくて見づらかったりしますから、そのときにフォントサイズを変えてやるのに使うといいかもしれません。とりあえず、配布されていた C 言語による復号化プログラムを載せておきます。

```
#include <stdio.h>
int main(){
  FILE *dat, *txt;
  int c;
```

```
  if((dat=fopen("nscript.dat", "rb")) == NULL)
    puts("nscript.dat が開けなかった"), exit(1);
  if((txt=fopen("0.txt", "wb")) == NULL)
    puts("0.txt が開けなかった"), exit(1);
  while((c=fgetc(dat)) != EOF) fputc(c ^ 0x84, txt);
  puts("終わった");
  return 0;
}
```

arc.sar(arc.nsa) は、立ち絵や背景などのデータが入っています。nscript.dat で書いたスクリプトから適宜こういったデータを呼び出して使用します。arc1.nsa,arc2.nsa... といったように多数のファイルに分割されていることもあります。nscript.dat はただのテキストデータなので容量はほとんど消費せず、ゲームで容量を占めるのはこの arc.sar(arc.nsa) と BGM、効果音になります。

あとは、ノベルゲーム用の True Type フォントを用意してやる必要があります。これは、実行するプラットフォームに関係なく Windows 用のものを使用します。好きなフォントを選べば良いのですが、フォントの種類によっては小さい文字が表示出来なかったりします。正しく表示出来るフォントとしては、みかちゃんフォント[1)]がおすすめです。

それでは、ONScripter を起動してみましょう。基本的にコマンドラインから起動することになりますが、Mac OS X では GUI のツールも配布されています。GUI のツールでも、中に CUI の ONScripter を内蔵していますので、基本は同じです。

```
# onscripter -r（ゲームのあるディレクトリ）-f（フォントのある場所）
```

で起動出来ます。いちいちフォントのある場所を指定してやるのは面倒なので、ゲームのあるディレクトリに一緒にいれてやるといいです。

後書き

なんといいますか、かなり微妙な仕上がりになってしまいました。もともとあまり文字数書けない話題だということは分かっていて、締め切りがかなりヤバかったのでまぁちょうどいいと思ったのですが、ちょっと短くなってしまいましたね。ということで、多少ページを埋めるために後ろにクイズみたいなのをつけてます。もっと早くから根回ししとけば割と深い別な内容を書けたような気もするのですが。今回は前回よりもさらに中身がない内容になった気がしますし。わりとネタは沢山あったので時間があれば良いものがもっと書けたような気がします。でも実質１日（！）しか時間がとれなかったのでしょうがない。まぁこれでも、スクリプトエンジンに興味を持って頂いて、Macintosh や携帯ゲーム機で Windows 用のノベルゲームを動作させることに興味をもってもらえる人が増えることを期待してます。

[1)] http://www001.upp.so-net.ne.jp/mikachan/

読み方問題

次の文字はなんと読むのでしょう。カタカナで答えてください。

Yahoo!	(	)
Aero	(	)
Adobe	(	)
Samsung	(	)
iTunes	(	)
Entourage	(	)
Mac OS X	(	)
Ethernet	(	)
Xeon	(	)
IEEE	(	)
SCSI	(	)

解答

Yahoo!	（ヤフー　　）
Aero	（エアロ　　）
Adobe	（アドビ　　）
Samsung	（サムスン　　）
iTunes	（アイチューンズ　　）
Entourage	（アントラージュ　　）
Mac OS X	（マックオーエステン）
Ethernet	（イーサネット　　）
Xeon	（ジーオン　　）
IEEE	（アイトリプルイー　）
SCSI	（スカジー　　）

・・・・2・・・・

Vista について

61 回生 浜田 克紀

前書き (という名の字数稼ぎ)

部室に Vista が導入されたのが今年の一月下旬で、買ったのは Business の 64bit 版。メモリを食うと聞いたので 8GB 積んだのだが多すぎたようです。

しかしほとんど誰も使ったことがない or 使えないので、XP と同じ感じでしか使わない。これでは Vista を導入した意味がなくなってしまう。結構色々問題も発生するし。てなわけで本まで買って頑張って使いこなす...とまではいかなくてもなんとか使っていける程度には技術を身につけたつもりです。てなわけで Vista のいろいろについて話させていただきますです、はい。

Vista を語る前に

この文章は話題の Windows Vista について書いたものですが、決して Vista すげーで終わる文章ではありません (別に Mac すげーともいいませんが)。否定的な面と肯定的な面を、他の OS と比較したり XP などの一世代前の OS との変更点などから見ていくつもりです。ここで紹介した技術を使う方法などは、Vista についているヘルプやグーグル先生[1]でちょっと調べたら出ると思うので書くつもりはあまりありません。

文中に時々出てくる Mac というのは、Macintosh の略です。今でこそ iPod や Macbook で apple が有名になったので知られているとは思いますが、つい 4 年前は自分も知らなかったので一応説明させていただきます。まあ自分が無知だったのはおいといて、これは apple という会社が開発した OS で、よく Windows と対比される OS です。ウィンドウズの実行ファイル[2]が動かないという点[3]以外は Windows でできる事はほぼ全く不自由せずできます。

後、コアな話は自分自身の技術と経験が足らないのであまり盛り込んでいません。あ、うちの部で使ってるのは正規品ですよ。

Vista とは

Microsoft が 2007 年上旬に、約 6 年ぶりに発表した、Windows NT シリーズの OS です。Vista の意味は、イタリア語で眺望、遠景という意味を表す言葉です。今までのメジャーバージョンアップは約 3,4 年間ずつの間を空けて発表していたのに Vista の発表がなぜこんなに遅れたのかを簡単に説明すると、XP で起こった色々な問題に対処し

[1] グーグル、あまりにも有名ですが一応 url を。`http://www.google.co.jp/`

[2] 拡張子が `.exe` となっているアレ、まあ Mac にはあまり拡張子という概念はありませんが。

[3] といっても頑張れば結構動くが、DirectX を使われていたりするとさすがに無理のようです。

ているうちに遅れてしまっい、これだけ遅れたからには Vista には多くの新機能をユーザーへ提供しなければならないというプレッシャーに苛まれてさらに遅れ、発表してみたらいつの間にか月日がこんなに流れていた、という感じです。

Vista の種類

Vista のエディションは、一般ユーザー向けの Home Basic、Home Premium、小企業向けの Business、新興市場向けの Starter Edition(すたーたーえでぃしょん)、大企業向けの Enter Prise(えんたーぷらいず)、そして家庭用かつビジネス用のヘビーユーザー向けの Ultimate(あるてぃめっと) の 6 種類があります。

具体的に利用できる機能は多すぎるので、大きな違いだけを言うと、Starter Edition は最低限機能するのに必要な機能のみ、Home Basic は家庭用の最低限の機能のみ (この二つは後述する aero が含まれていないのが大きな違い)、Home Premium は家庭用で使うと思われる機能 (動画関連のソフトが入っている代わりにバックアップや暗号化などの機能が含まれていない) を盛り込んだもの、Business 及び Enter Prise は仕事で使われるような機能 (動画関連と後述する Bit locker というもの以外のほぼ全ての機能、Enter Prise はさらに OS の仮想サーバーなどのソフトがついてくる)、そして先に述べた Ultimate は全ての機能をそろえています。

機能の多さ及び値段は Starter<Basic<Premium<Business<Enter Prise≤Ultimate の順です。

また、一般ユーザーの購入形態は DSP 版や OEM 版というのが存在し、DSP 版は Microsoft(以下 MS) でなく店側がパソコン自作ユーザー向けに設定したもので、PC パーツと共に買うと OS の値段が格段に安くなる[4]という仕組みで、そのパーツと共に使用することが条件である。OEM 版というのはメーカー製のパソコンに始めから OS が入っているもので、そのパソコンと共に使用することが条件である。こちらの場合 OS の DVD が付属してこないことが多々あり、一般ユーザーは意識せずにこちらを使っている事が多い。

そして最後に 32bit 版と 64bit 版についてですが、これを語ると結構長くなるので簡潔に話すと、32bit 版は 2000 や XP の大体のソフトが普通に動く代わりに 3GB 以上のメモリを読み込めないという欠点がある。なぜかというと、1GB=2^{10}MB=2^{20}KB=2^{30}B というもので、32bit というものは 2^{32}bite の情報を扱えるものであるので、4GB までしか無理であり、さらに 32bitOS の制限[5]により 1GB 食われるので 3GB までしか読み込めないという事態になっている。そして 64bit 版。これは 2^{64}bite までの情報を扱える。さっきの 2^{32} と比べて 2^{32} 倍の情報が扱える。正直制限などないに等しい。しかし互換性の関係で昔のソフトが開けなかったりする。最近は色々環境も整いつつある。今は 32bit → 64bit の過渡期なのでどちらにするかは個人の判断ですが、普通のユーザーなら 32bit が無難だと思われる。

---

[4] 原価は一万円もしないくせに、4,5 万で通常販売するというありえないぼったくりに比べればまし、という程度。それでも 1 万 5 千円〜2 万 5 千円くらいはする。

[5] 32bit により使えるメモリが 4GB まで制限を受けるとはいいましたが、すべてのメモリの合計が 4GB という意味です。たとえばグラフィックボードのビデオメモリが 512MB だとすると、残りの 3.5GB しか通常の用途には使えません。Windows ではユーザーのことを思ってのことでしょうがメインメモリの要領の合計を 3GB までと制限しています。ですからもちろん色々パソコンに刺して、そのメモリの合計が 1GB 以上の容量になると 3GB よりも減ってしまいます。

ちなみに Mac OS X(テン) 10.3 以降では、64bit 化の問題が起こらない様子。余談だが、あの有名な Nintendo 64 は 64bit だから 64 なのです。そんな昔にあったのかと驚かれるかもしれないが、ゲーム専用機なのでそこまで難しいことはないのです。

結論としては普通の方は Home Premium を使えばいいかと思われます。新しく買うパソコンには大体これが入っているはずです。アップグレードするときに Home Basic の方がほんの少し安いからといってそっちを買ってしまうと幻滅するかもしれません。個人的には、最低限の機能だけを使いたいのならむしろより軽い XP を使ったほうがいいかなと考えています。

少しアングラな話かもしれませんが、DSP 版をインストールしているときにエディションの選択という場面があり、Home Basic、Home Premium、Business、Ultimate の 4 つの選択肢が出てくるので、もしやと思って調べてみたところ、案の定これらのディスクの中身は同じでした。(32bit 版と 64bit 版は違いますが。)

実はインストール CD の表に貼ってあるシリアルナンバーという番号でこの 4 つのエディションを振り分けているだけで、ディスクの中には Ultimate、つまり全ての機能が内包されているようです。

といってみたのですが、実は XP では回避可能です。設定を変えて、ユーザーが手動で振り分けてやることが可能ですので。しかし Vista では無理です。物理的には可能ですので、本気で色々と頑張ったらいけるのかもしれませんがとんでもない労力と時間を食うと思いますし、壊してしまう危険性も伴うと思いますのであまりお勧めできません。

XP **にない** Vista **の新機能**

●● Aero(**あえろじゃないよ、えあろだよ**) ●●

今 Vista を買うとなんとチョコレート菓子とトイプードルが付いてきま...せん。

これは Vista で新しくついてきたグラフィックボードを活用する技術の名称で、Premium 以上についてくるものです。

XP 時代と比べてとてつもなく進化しています (XP → Vista の発表に 6 年もかかったんだから進化しなければ逆に怒りますが)。具体的にはウィンドウ半透明化、動的ウィンドウ、タスクバーにマウスポインタをあてたときのミニ画面の表示、ウィンドウズフリップというウィンドウのサムネイルを `Alt`+`Tab` で表示する機能、ウィンドウズ フリップ 3D というウィンドウズキー+`Tab` で 3D 表示することによりウィンドウの切り替えがスムーズかつ綺麗に行える技術等があります。

しかし、これらの機能がちゃんと使えるのはまあまあのグラフィックボードのみなので、昔からのものでは残念ながら実行できない可能性が高いです。

一方、この Aero と似ている (というか Mac 信者からは Aero はこれのパクリだとか言われていたり) ものとして Mac の Aqua というものがあります。自分は Windows ユーザーなのであまり詳しくはありませんが、Mac OS X から採用されたもので、少し使ってみた感じは Aero とかなり似ているなぁという第一印象を持ちました。

まあ使えるなら当然使ったほうがいいですが、グラボが弱いとあまり効果を発揮しないという感じの機能です。

詳しく説明すると、XP の時代は処理をすべて CPU に任せていたのですが、Aero はグラフィックボードを効率よく利用することができるので、Aero を利用したほうが軽くなるというわけです。

●● クイック検索 ●●

あなたは XP もしくはそれ以前の Windows の検索をご存知ですか？あれのヘルプには検索で探したいファイルが見つかったなら検索を切ってくださいと書いてある。つまり遅いと認めているということだ。ためしに自分のパソコンの 60GB の HDD でファイルとフォルダの一部に”npca”と入力して検索したところ、いるかくんが頑張って 4 分かけて検索してくれました。

...口調が変ですね。とりあえず XP の検索はユーザーを大分いらだたせます。

まあこのありえない速さが Vista になってかなり改善されました。Vista のクイック検索は、インデックスという、本で言うところの索引のようなものをスキャンすることにより、ゼロから検索する XP に比べて格段に速いというしくみになっています。ちなみにスポットライト[6)]は大分速くて、普通に開くのが面倒なときに名前を入れて一覧を出すとか言う恐ろしいことをこの前やってのけていた。というのは、Mac の期限である UNIX は、検索が速くなるように始めから作られているからである。まあこれは仕方ない。

●● Readyboost ●●

Vista はメモリを食います。一応無駄にメモリを食ってるわけではないのですが (後述する Super fetch の影響もあるし)、それを差し引いてもメモリをかなり食います。新しいマシンなら問題ないと思いますが、古いマシンだとメモリをあまり詰めない可能性もあります。そこで、Vista では USB フラッシュメモリを通常のメモリとして利用することのできる、Readyboost という機能が組み込まれています。これには速さと容量の条件があるものの、結構使える機能ではあります。ちなみに一つまでしか使えません。

この機能、かなりメモリの速さが操作性に影響します。とりあえず速めのフラッシュメモリで、本体とあわせて 2GB 以上の容量になるのが望ましいようです。

ですが、あくまでメモリをこれ以上させないパソコンのための非常手段なので、性能にはあまり期待しないほうがいいかもしれません。後、もちろん 32bit ならば 3GB の制限を受けます。

●● Readydrive ●●

この機能はハイブリッド HDD を使うときの機能です。のでまずハイブリッド HDD について。

最近のパソコンは速いですよね。CPU とかクアッドコアとか出てるし、メモリも積もうと思えば 16GB(は一般的じゃないけど、普通にやろうと思えば 8GB はもはや現実のものとなった) も積めるし、グラボも普通のを使うだけで通常のユーザーなら不便だと思うことはもはやないと言っても過言でない時代になってきました。どうでもいいですけど Windows 2000 までではクアッドコアを認識しないようです。そりゃあの時代に出てなかったから当たり前ですが。

しかしここでネックとなってくるのが HDD。磁気にガガガと書き込むのを遅く感じている人は少なくないはず。ならメモリを使えば...と思うかもしれませんが、単価が高すぎます。現在だと 25〜40 倍くらい？の値段がするはずです。値段がばらばらすぎますが、とにかく大容量を使うには HDD は必須です。で、どうすればいいのかというと、簡単に言うとメモリと HDD のいいとこどりをすればいいのです。

[6)] マックにおける検索システムの名称

HDD は一つのファイルを読み込んだり書き込んだりするのなら、少しヘッド[7]を動かしてファイルをいじればいいので電位でデータを保存しているフラッシュメモリよりも速いのですが、ランダムにヘッドを動かすのは何度も何度もくるくると円盤をまわさないといけないのでとても非効率で遅いのでフラッシュメモリのほうが断然速いのです。

詳しく説明...するもなにも HDD にちょっとフラッシュメモリをとりつけてみただけのものです。これでどうなるかといいますと、いちいち HDD にアクセスしなくてもよくなるので高速化 (但したまに普通の HDD より遅い)、OS の高速起動 (OS の起動には HDD のランダムシークが必要でめちゃくちゃ時間がかかるのに対してフラッシュメモリはそれが大幅に短縮され、また電源を切っても内容が保持されるから)、省電力 (結構 HDD は電気を食う、むしろこれがハイブリッド HDD のノートパソコン内での利点でもある)、静音性 (これはノート、デスク問わずいい感じ)、耐久性 (そもそも寿命としては HDD¡¡¡メモリで、HDD の読み書きの回数を少なくすることで寿命を延ばすことができる)、熱対策 (HDD ってパソコン内で結構熱い部品だからねぇ)、耐衝撃性 (ヘッドは書き込むときはプラッタとの距離がとてつもなく近いので、書き込んでいるときに衝撃を受けるとプラッタと接触してしまい、データの書いてある部分が傷ついて、結果 HDD が壊れてしまうこともあるので、これを避けるために、外部衝撃がほぼ無効のメモリに極力書き込むことで HDD を守る) などがあります。

んで、当初の Readydrive ですが、これは Vista がハイブリッド HDD のメモリを効率的に利用することができる技術で、普通の HDD よりもさらに速くすることができるものです。

普段は書き込まないとき退避エリアという安全圏にいて、データを記録してある円盤との接触を絶対にさせないようにしてあります。

●● Superfetch ●●

主婦というのはスーパーでの買い物が好きなのです。これをスーパーフェチというのです。…というのは真っ赤な嘘で、Superfetch というものは、ユーザーが使うアプリケーションの頻度を OS 側が監視しておき、頻度の高いものをあらかじめメモリに読み込んでおくことで従来の起動した瞬間から読みこむ方式よりも高速化を計る技術です。

Vista がメモリを食うといわれていますが、このような仕組みでメモリを最大限活用しているから一見するとメモリを食っているように見えるだけなのです...と Vista 擁護派はいうが、実際に Vista は結構重くなっています。時代が進んだからというのもありますが、ユーザーが今求めているのは機能よりもむしろ軽さになってきているようなので、そこら辺を考慮して作ってもらえる事を期待しています。

●● Windows Complete PC、シャドウコピー ●●

Vista では自動システム回復機能より優れた Complete PC というものが用意されています。(でも business 以上にしかないようですけど)

主な違いは、XP の自動システム回復機能にはバックアップ機能の制御を行う機能はありませんが、これにはあります。

そして、操作性が格段に上がっています。XP が無駄にやりずらかっただけという表現が正しいと思いますが...

---

[7] ハードディスク内部で、プラッタという部分にデータを書き込むためのもので、プラッタとの接触はせず、ナノメートル単位で離されています。

以前は一部の情報をフロッピーにごちゃごちゃと書かないといけなかったのですが、こちらはイメージという、一つのファイルでバックアップできるようになっています。

また、HDD でなく DVD でもいいようです。(もちろんもとの HDD の容量分の枚数が必要ですが)

そしてシャドウコピーというのは、任意のファイルをユーザーがいちいちバックアップしなくても、裏で勝手にバックアップしてくれている (というより前回との差分を記録している) 機能です。

これでうっかりファイルを上書きしても大丈夫です。但し消した場合は無理 (のはず) です。

内部的な違い

●● ブートローダーの違い ●●

Vista と 2000、XP などとはブートローダーが違います。ブートローダーというのは、OS を起動するときに最初に起動するものです。これが違うので、Vista の入ってるパソコンに XP や 2000 を普通にインストールすることはできません。詳しく説明するとかなり長くなるので省略しますが色々細工が必要です。

●● 細かい違い ●●

後、32bit ではありませんが、64bit では program files が二つあります。どういうことかというと、32bit 用のソフトを入れるための”Program Files”というフォルダと 64bit を入れるための”Program Files(x86)”というフォルダです。

とりあえず全体的な結論として XP → Vista に移行すること事態は別にかまわないと思いますが、その際にパソコンを買い換えるなり部品を組み買えるなりしないと、Vista の新機能がうまく働かなかったり、むしろ負荷が裏目に出て XP の方がいいという結果になってしまうこともあります。

具体的には OS 自身にあてるメモリ不足 (Readyboost で対応できることもある)、グラボが弱いため aero がうまく動いてくれない (これはグラボをかえるか Aero を切るかで対処でき…るのか？)、

個人的には微妙なマシンなら 2000 が最高だと思っている、大分軽いし頑張ろうとしている XP のインターフェースよりは 2000 のシンプルな奴のほうがむしろ綺麗に見えるかもしれませんし。唯一まずいな点が Microsoft がサポートする気をなくした点だけでしょう。しかしもしそれで不自由に感じるようになってきたとしたら、Vista に買い換えればいいことでしょうし。

まあ今が買い替え時かどうかは知りませんが、2009 年に新しい OS が出る予定も念頭において買うかどうかを決めるといいと思います。

あとがき

5,6 時間で書いたこの適当な文章をお読みいただき本当にありがとうございました。

ちなみに私自身のマシンは古すぎて、Vista がちゃんと動く気がしません。なのに何故この部誌を書いた (書けた) のでしょうねぇ？本当に世の中不思議なことだらけです。

とふざけたことを言いたいわけではなく、Vista はまだまだ改善するべき課題をかな

り残しています。それに OS というものは、ハードウェアが時代とともに進化していくのでそれに伴う改善が必要なので完璧な OS というのはないと思います。

現在、Vista Service pack 1 という機能改善版が発表され、メーカー製のパソコンにも導入されてきている模様です。今から買うとおそらく OS は SP1 となるのではないのでしょうか？

なお、ここで紹介した事は、Vista についてのごく一部です。興味があったら調べてみてはいかがでしょうか？

では、ごきげんよう。

・・・・3・・・・

幾何学・アルゴリズム体操！ 目と指の！

63 回生 佐野良輔

いつしか部誌を書かなければならない学年と季節がやってきて、過ぎて行きつつある。パクリである。とはいえ、自分は他の先輩方に比べると PC、及びプログラミングの知識については無に等しく、パソコン部にいて果たしていいのだろうかと疑問に思ったりもする。しかしまあ、書かなければいけないのは事実。部員の少なさもあって、このままでは今年の 10 月頃に部長になってしまうのだ。このままでは最悪。

プログラミングの知識についてはこれから半年かけて伸ばしていくとして、今眼前に立ちはだかる強敵、むしろ無理敵ぐらいの威圧感をかもし出している、「部誌」という名の敵。これを残る 4 時間で倒さなければならないのだ[1)]。つまり何が言いたいかというと『字数を稼ぐ必要がある』わけで。そうはいっても内容がなければ始まらないのでさっさと始めることにするけれども、これから説明するのは自分の唯一の、あえて挙げろと言われたら特技と言ってもいい、「ゲーム」。具体的に言うと「弾幕シューティングにおける弾幕のアルゴリズム」、あとアルゴリズムそのもの。

何を言っているのかわからない人は次を読んでもらえればわかると思う。

アルゴリズムの話

例を挙げれば、素因数分解。

素因数分解のやり方といえば、分解したい数の二乗根の数値を超えない素数までででひたすらその数を割っていくことだが、これはどの自然数を代入してもやり方そのものは同じだ。つまり、やり方を予め決めておいて、そうすれば数値によって結果を変えられる、ある意味方程式のようなものだと思えばわかりやすいだろう。$X + Y = 1$ なんてものも方程式だから、もちろん代入する変数はひとつではない。弾幕なら、弾の速さ、角度、大きさ、などなど。

まあ詳しくは、次から見ていくことにするが、その前にひとつ謝っておかなければならないことが。

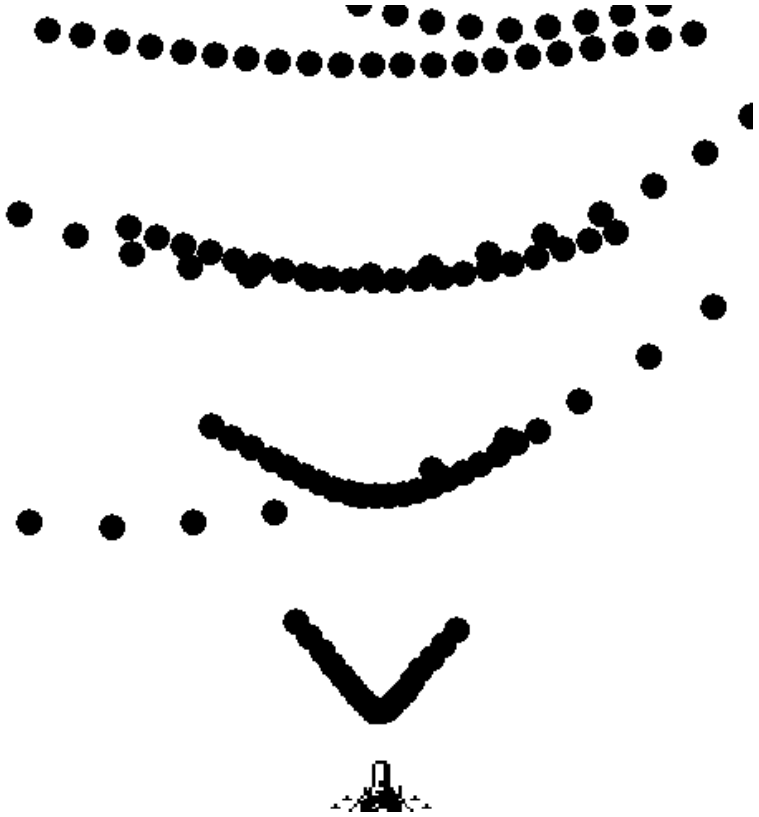
タイトルはどうみても適当です、本当にありがとうございました。

弾の種類の話

意味不明で申し訳ない。が、まずはこれを見ていただこう。俗に言う (のかは知らないが)「弾幕」の一種・自機狙いである。先輩による一昨年の文化祭作品から。

[1)] 4 時間では倒せませんでした

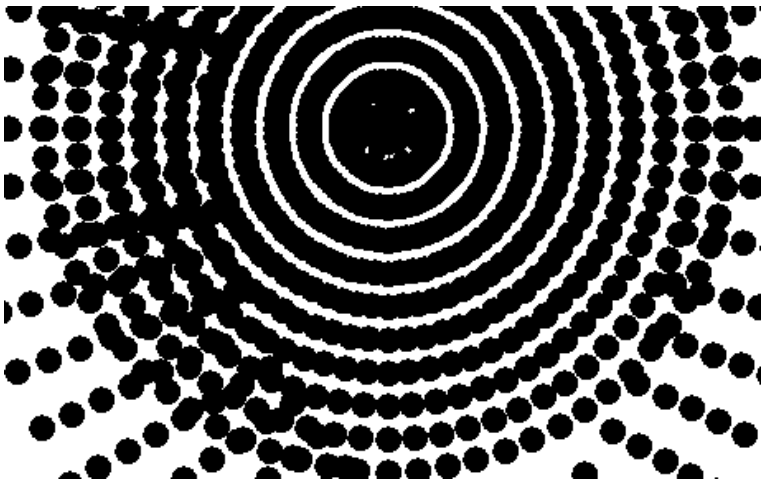
●図 3.1 自機狙い弾



自機に向かって色々と飛んできている。その名の通り、自機狙い。速度は同じだが距離が違うので、列になって飛んできている。

自機を狙うのだから、自機の座標の情報は必要だ。弾の発射地点と自機との座標を見比べて、その差をとれば方向ベクトルがわかる。そうなればあとは適当に速度を合わせるように関数に放り込むだけだ。言うほど簡単ではないが、言うほど難しくもない。

●図 3.2 72way 弾



中央の敵がやたらめったら弾を出している。全方向 72way 弾というところか。

全方向に弾を一定の角度を持って出す場合、必要なのは弾の速度と方向だ。「大きさ」と「方向」を持つ数値…というわけで、再びベクトルの出番である。

というのは嘘で、

別にそれでやってもいいんだが芸がないというか他の方法で考えた方がわかりやすいのでここは円らしく角度 (ラジアン) を使うことにしよう。

この場合 72 方向に弾が出ている。一周は 2π ラジアンなので、$\pi/36$ ラジアンごとに弾を出せば万事解決。右方向を 0 度、角度を表す変数を r(最初は 0) とする。弾を撃つ関数の引数として $0, \pi/36, 2\pi/36 \ldots$ としていくわけだが、この場合すべての弾を同じ速度にしようと思うと、速度 1 ごとに弾が進む x 座標を $\cos r$、y 座標を $\sin r$ とすればいい。

※もう少しだけわかりやすく

$$\sin^2 r + \cos^2 r = 1$$

つまり、どの角度でも中心から同じ速度で離れていくことになる。これに速度をかける。ちなみに角度によって速度を変えて何回か連続で全方位弾を出すと、時間差で飛んでくる弾幕も作れる。途中で速度を変えるようにするのも面白い。[2)]何フレームかごとに 1 速度を足していくような[3)]ことにすれば、加速弾なんかも作れてお得。

ちなみに全方向弾を作る際、必ずしも 360 の約数でないといけないというわけではない。後で説明もするが、自機の動きを制限するいわゆるワインダーなんかは全方向弾でも自機が入る部分だけ広めに取ってあったりする。その場合は、自機の方向+5 度ぐらいを 0 として適当にいいところまで足していくのだが。

[2)] フレームで速度を変えればいい

[3)] つまり、弾が発射されてからのフレーム数が、例えば 5 で割ると 2 余る数だとかの条件分岐を使う

バカとハサミの使い方の話

アルゴリズム、とは前項で紹介した自機狙いやら全方位弾やらの１つ１つなのだが、しかし弾幕というのはこれらを組み合わせてなるものだ。その組み合わせ方で難易度も変わってくる[4)]。

アルゴリズムの便利なところは、一度それを作ってさえおけば、数値を変えることでいくらでも弾幕なら難易度を変化させることができるから、というのが一番大きいだろう。それに、自機狙いを作る→レーザーを作る→全方位弾を作る、ときて、この３つの内いくらでも組み合わせられる。取捨選択が可能なのだ。

例)

1. 自機基準の全方向 72way 弾
2. 弾の発射源を動かす

自機基準の 72way については、`atan2()` 関数を使えばいい。座標の差を入力すると勝手に角度を計算してくれる優れもの。あとはさっきのラジアンの要領で。[5)]発射源を動かすのは、フレームごとに勝手に動くようにしていればいい。

簡単に作れる弾同士の組み合わせでも、面白い弾幕が作れたりする。他のゲームの弾幕を作ることに楽しみを感じる人がいるのは、そのせいかもしれない。

オリジナリティの話

動きがどうみても単純なものではない弾だってもちろん存在する、というよりいくらでも作れる。加速度をランダムにして n フレーム後に分裂、とかやるだけでも、単にうざったいだけだろうが、オリジナルの弾幕だ。

ところで、分裂について説明してなかったことを今更思い出した。といっても単純なもので、弾を消すと同時に弾を出せばいいのだが。その際弾の座標が発射源の座標になる。

数学の知識が必要なことは、確か。数学の勉強をしていれば面白い軌道の関数だってお目にかかるかもしれないし、美しい幾何学模様、というものも知れる。そんな「作ってみたい模様」を関数で表記し弾の軌道にして際限無く弾を出し続ければ、とても綺麗な幾何学模様が不思議と弾幕によって浮かんでくるわけだ。

関数同士を組み合わせることによって、幾何学を完成させるアルゴリズム。やっぱり、プログラミングやらが面白いと思う瞬間でもある。

まだ体験してないけど。

言い訳の書き方

終わってしまった。

いうまでも無い、ネタ切れだ。そもそもなぜレーザーも作れないのにこんなものを書こうなどと思ったのか、今でもさっぱりである。

ちなみに世間には誘導レーザーなんて奇妙なものもあったりして、作れるならばそれの紹介で１ページは消費していたであろう。いかにもややこしそうだし。

[4)] たまに完全なオリジナルの軌道を持つ弾幕があったりするが

[5)] さっきの自機狙いもそうすればよかったんじゃないかって?処理重いじゃないですか。

弾とレーザーの一番の違いといえば、「くぐれるかくぐれないか」だというのは間違いないと思う。上で述べたワインダーなんかも、人間技ではないが抜ける余地はある。たまにない。

なぜ後書きでワインダーの説明なんかをしようとしているのかというと、単純に忘れてたから。つくづくダメ人間だ、最近はやりの NEET にはならないよう最大限の努力をするつもりではあるが、果たして。

そういえばひとつ勘違いをしていたのは、NEET というその言葉自体には本来は、「働きたくない人」という意味はないそうで。「教育を受けていない、労働をしていない、職業訓練も受けていない」というわけで。どうでもいいや。

あ、ワインダーは「自機の動きを制限するような、でも完全には制限できてないものすごく速い弾」だと思ってくれればいいです。作り方も自機 ±5 度に打つとかでおーけー。

なぜ後書きになってこうまで文体が崩れているのか説明すると、疲れたから。あと時間がないから。

とまあ、あらかた思いつく言い訳と理由を書いてきたけれども、果たしてこれが後書きになるのかそれはわからない。ただ単に書きたいこと、書ける最大限のことを書くように努めただけだ。

最後はうまく締まったと思うが、どうだろう？

来年は、これの 2 倍の分量は書きたい。と思う。自信はない。

ゲームが完成していることを祈りながら、先輩方にバトンタッチ。

・・・・4・・・・

はじめての C

62 回生 平野湧一郎

C 言語とは

皆さんは「C 言語」という言葉を聞いたことがありますか？コンピュータは 1 と 0 で書かれた命令を理解し、実行しますが、生憎私たちは人間ですので 1 と 0 で書かれていても何の事やらわかりません。そこで、人間が分かるような言語 (高級言語) でプログラムを書き、これを機械が理解できる言語に翻訳することでプログラミングをします[1)]。そのような言語の一つが「C」という言語なわけです。

C は 1972 年に発明され、以後世界中で非常に広く使われています。35 年以上経った今でも、最もポピュラーな言語の一つと認識されています。その理由としては、言語仕様が適度にコンパクトなこと、それにも関わらず非常にパワフルなコーディングができることなどが挙げられます。しかも、C 以後に作られたほとんどの言語 (全てと言ってもいいでしょう) は、多かれ少なかれ C の影響を受けているのです。

また、プログラミングの大会でも、特定言語でのプログラムを競うことが目的でない場合、ほとんどの場合 C を使えないことはありえない、といっても過言ではありません。前述した通り、C 以後の言語はほぼ全部が C の影響を受けているので、C を使えることは非常に強力な武器になるのです。

それでなくても、プログラミングすることはとても面白いものです。

ここでは、C 言語の基礎をご紹介したいと思います。

基本プログラム

C 言語の基礎は、例えば次のようなプログラムです。拡張子を「*.c」にして、テキストエディタで作ってみましょう。「はじめての C」以外は、アルファベットは全部小文字で書いてください。

```
#include <stdio.h>
int main()
{
    printf("はじめての C¥n");
    return 0;
}
```

このプログラムが何を出力するか、分かりますか？なんとなく、「はじめての C」と出てくるような気がしますよね。

---

1) ちなみに、この翻訳作業を「コンパイル」といいます

実際には、このプログラムを実行するためには「翻訳」作業が必要となるので、これを書くだけでは実行できないのですが、それについては後で詳しく説明します。

解説

では、このコードの解説をします。

`#include <stdio.h>` いきなり難しい記述が出てきましたが、これはおまじないです。C のプログラムを書くときは、最初の行にこれを記述するもの、と覚えてください。

実際は、「`stdio`」は「`standard input & output`」の略で、「標準入出力」を表しています。簡単に説明すると、画面に出力したり、画面から入力を受け取ったりとかいう処理が「`stdio.h`」というファイルにまとめてあって、それを読み込んでいるということです。

ちなみに、プログラムを作る毎にこれを書くので、慣れてくるとタイプするのがやたら速くなります。

`int main()` “main”、つまり「本体」です。プログラムの本体はここから始まりますよー、ということを表します。カッコの中には何も入ってませんが、省略することはできません。やはりおまじないだと思ってください。

ところで、「`int`」というのが出てきましたが、これは「integer」、つまり「整数」を意味します。どこが整数か・・・というと、「`return 0`」の「`0`」が整数ってことなんですね。って言っても良く分かりませんが。

{ 開き中カッコです。`main` が始まることを意味しています。後の”`}`”(閉じ中カッコ)と対をなします。

「なんで前後で改行するの？」と思うかもしれません。実は、以下のように改行無しで書くことも可能です。[2)]

```
#include <stdio.h>
int main(){printf("はじめての C¥n");return 0;}
```

ただ、このようなコードをずらずらーっと書いていったら、「`main` はどこから始まるんだっけ？」ということが起こるわけです。カッコに一行使えば、`main` が始まることが一目瞭然ですね。つまり、見やすさのためにわざわざ一行使ってカッコを書いているのです。

「カッコで一行使うなんて気持ち悪いよー」という人はずらずらーっと書いていっても構いません。ですが、できるだけ見やすくした方がいいでしょう。

`printf("はじめての C¥n");` 「はじめての C」という文章を画面に表示します。`""` の中の文字列を他のものに変えてやれば、それを表示します。

なお、「`¥n`」はこれ一文字で改行を表します。PC によっては、「`¥`」ではなく「\」(バックスラッシュ) が表示されることもありますが、同じ意味です。

[2)] 最初の一行だけは個別に一行必要。

左にスペースが少し空いていますが、これもやはり見やすくするためです。このように、左のほうを少し空けてプログラムを見やすくすることを「インデント」といいます。

インデントは大抵、「Tab」が使われるか、スペース 2〜4 個で行われることが多いです。最後に「;」(セミコロン) が付いていることに注意してください。良く忘れます。

`return 0;` "return"は「戻る」、つまり「ここで `main` は終わりです」ということを表します。なんで 0 がくっついているのかは・・・まあ気にしないでください。別に 1 でも 100 でもいいですが、伝統的に 0 が使われているだけです。

} 前述した開き中カッコと対応する、閉じ中カッコです。

どうですか？意外と簡単だと思いませんか？簡単なプログラムなら、たったの 5 行で書けてしまうのです。

もちろん、より高度なことをするには、より難しいプログラムを書かないといけません。しかし、プログラムというのは基本、簡単なことの積み重ねなのです。

あるいは、たった「はじめての C」と表示させるだけなのに 5 行も使うのは、結構難解に思えるかもしれませんが、書いてみればすぐに慣れます。始めは戸惑うかもしれませんが、書いてしまえばどうということはありません。早速、テキストエディタでも何でもいいので、好きな文字列を表示させるプログラムを作ってみてください。

実行してみる

さて、プログラムは書けましたか？今書いたプログラムはあくまで「ソース」(人間が理解できる言葉で書かれたプログラム) ですので、これをコンピュータが理解できる言葉[3]に翻訳してやらなければいけません。

前述したように、この作業を「コンパイル」と呼びます。また、コンパイルしてくれる奴[4]のことを「コンパイラ」といいます。

C のコンパイラで最も有名なのは、「Microsoft Visual Studio」でしょう。これは C だけでなく、C++だろうと Windows API だろうとコンパイルしてくれる上に、非常に使いやすいのでお勧めなのですが、有料です。パソコン部の PC には全て入っていますので、パソコン部に来て頂ければ使うことができます。自宅でプログラムを組みたい方や、ちょっとお試しでプログラミングをしてみようかな、という方には、フリーのコンパイラをお勧めしておきましょう。

フリーコンパイラで最も有名なのは、「Borland C++ Compiler」でしょうか。ユーザー登録が必要ですが、登録したメールアドレスに滝のごとくメールが雪崩れ込んでくる、ということはありません。ですが、インストール後もいくつかしなければならないことがあり、設定がちょっと面倒です。インストール方法や使用方法は割愛しますので、ネットで調べてください。URL を載せておきます。

`http://www.codegear.com/jp/downloads/free/cppbuilder`

インストールできたら、早速コンパイル・実行してみましょう。ファイルが例えば `C:¥hogehoge.c` なら、コマンドライン (∗) で次のように打ち込みます。[5]

---

[3] 平たく言うと、0 と 1 からなるプログラム。といっても、このソース自体が実は「0 と 1 からなる」のですが、その辺はややこしいので省略します

[4] 実はこれも、何らかの方法でコンパイルされたプログラムなのですが...

[5] ちなみに、Microsoft Visual Studio なら、コードを打って F7 を押すだけでコンパイルしてくれるので、とても簡単です。

```
bcc32 C:¥hogehoge.c
```

これでコンパイルされます。文法に間違いがあると「error:(**エラー内容**)」などと表示されるので、表示されたエラー内容を参考に、気合でエラーを消してください。「warning」と表示されることがありますが、これはコンパイラが「警告」しているだけなので、実行することは可能です。

エラーがなくコンパイルが終わると、hogehoge.c と同じフォルダに実行ファイル (多分 hogehoge.exe) ができるので、実行してみましょう。「はじめての C」と表示されるはずです。

プログラムを実行してみて、どうでしょう？コンピュータが自分の書いた命令を実行しているのです。なんだかとっても楽しいですよね。このプログラムは最も簡単な例ですが、もっと詳しいことが知りたい場合、ネットで調べてもいいですが、それよりもパソコン部に来て下さい。親切に教えますよ!

FAQ

Q：FAQ ってなに？

A：Frequently Asked Question(頻繁に聞かれる質問) の略です。~~要するにページ稼ぎ~~

Q：C を極めれば、どんなことが出来るようになるの？

A：パソコン部の文化祭に来てください。といっても、この部誌を読んでいるということは文化祭に来ているということでしょうが。

Q：プログラミングに英語の能力は関係ありますか？

A：基本的な単語の意味ぐらいは分かった方がいいと思います。「return」とかね。
ですが、そこまで英語が出来なくてもやっていけると思います。難しい単語を多用するとソースが読みにくくなりますし、プログラミングによく使う単語は、受験英語ではあまり重要でない単語も多いですから。
どちらかというと、いい名前をつけるセンスや、国語力・論理力といったものが重要だと思います。

Q：プログラミングに数学の能力は関係ありますか？

A：基本的なプログラムを書く際は、四則演算が出来れば十分だと思います。数学のプログラムを書く場合でも、三角関数やベクトルぐらいが理解できればＯＫです。
なお、数学的思考力や美的センスはきわめて重要です。がまあ、プログラムを多く書くうちに備わってくると思います。

Q：シューティングゲームみたいな、3D で動くカッコイイゲームが作りたいのですが。

A：C は「万能言語」ですから、Visual C++や DirectX といったものを用いれば、勿論そのようなゲームも作ることは可能です。
ですが、コンソール (黒画面) で動くプログラムに比べると、C でそのようなプログラムを作るのは物凄く大変なことなのです。少なく見積もっても、半年ぐらいは「修行」を積む必要があると思います。
もしあなたが「手軽にカッコイイゲームを作りたい」と思っているのなら、別に C に固執しなくても、Visual Basic や RPG ツクールといった言語を採用するほうがいいでしょう。
ですが、やろうと思えばほとんど何でも出来る C に比べると、今挙げた言語—さらに言えば、手軽にカッコイイゲームが作れる言語全て—は、どんなに頑張っても

実現不可能なことが多いです。
ですので、自分がやりたいことを全て実現したいなら、多少回り道でも C を学ぶことをお勧めします。
それに、C 一本でもプログラミングの大会にはほとんど全て出場することが出来ますし、大学受験や就職も多少有利になる・・・かもしれません。

Q：プログラミングの大会って何するんですか？

A：「プログラミングの大会に出て・・・」とか言うとほぼ聞かれる質問です。人間の手では計算不可能な問題が一つ出ます。[6]それを解くプログラムを作り、実行させるわけですが・・・まあね、私程度のプログラミング能力だと、「どうすれば求まるのか分からない」とか、「プログラムは書けたが、実行しても終わらない」ということがそれはそれはよくあるのです。
例えばデータ数が一万個なら、「実行時間がデータ数に比例するプログラム[7]」は、「実行時間がデータ数の二乗に比例するプログラム[8]」の一万倍速く終わるのです。これは決して誇張ではなく、問題によっては模範プログラムと私の書くようなドヘタプログラムで一億倍の速度差が出ることも珍しくないのです。
ですから、プログラミング大会では、「答えを求めるプログラムを書くこと」と同時に、「できるだけ早い時間で終わること」、例えばデータ数を N としたとき、実行時間が N や $N \log N$ に比例するプログラムを書くことが求められているといえます。
プログラミングの大会に興味があるなら、スパコンや情報オリンピックの予選に出場してみるといいでしょう。

参考 URL

参考になるかもしれない URL を載せておきます。~~要するにページ稼ぎ~~

Borland C++ Builder `http://www.codegear.com/jp/downloads/free/cppbuilder`
フリーのコンパイラです。コマンドラインを使ってコンパイルします。使いやすさでは Microsoft Visual Studio に劣りますが、無料です。

Microsoft Visual Studio `http://www.microsoft.com/japan/msdn/vstudio/`
C、C++コンパイラの金字塔。有料です。学生なら少し安く買えます。

WisdomSoft `http://wisdom.sakura.ne.jp/programming/c/index.html`
とっても有名な言語解説ページです。ところどころかなり難しいですが、C のマスターを目指すなら一度は読んでおきましょう。
個人的には猫より分かりやすいと思います。

DirectX `http://www.microsoft.com/japan/windows/directx/default.mspx`
私のようなめんどうくさがりなプログラマのために、描画とかを担当してくれるソフト (?) です。

---

[6] 例:町の地図と、地点と地点の間の移動時間が与えられたとき、A から B までの最短時間と、それを与える経路を求めよ。

[7] $O(N)$ と書きます

[8] $O(N^2)$ と書きます

所謂「動くゲーム」を作るときはとってもお世話になります。

情報オリンピック日本委員会　`http://www.ioi-jp.org/`

orz

それでは、これでおしまい。

あと、この文章を読んだ新入生の方、旧校舎 4F のパソコン部部室に見学に来てください。マジでお願いします。

・・・・5・・・・

おわりに

部長 浜田克紀

どうも、前部長の katukky です。

二年連続でこの部誌を読んでくれている奇特な方がいらっしゃったならば、すでにお気づきかもしれませんが、前書きを見比べてみてください。ほんと、手抜きとかいうレベルではありませんね。

さて、部誌のあとがきなのですがたいしたことも書ける気がしませんので、今の自分の心情でも語っておきます。

文化祭が終われば、僕はほとんどパソコンに触る機会を失うだろう。その前に心残りがないように色々な事をしておこうと思っている。この部誌を描くのもその一環であったりする。

そして印刷前の原稿を見てみると、高 3 が部誌の過半数を占めている状態に気づく。このままでは、来年はこの部誌の存在すらも危ういのではないかと本気で思っている。まったくこんな事をいえるような立場でもないはずなのだが、本音だから許してほしい。

それでも希望の光もある。中 1 達だ。現在は中 2 という細かい事は置いといて、パソコンスキル及びプログラミングスキルを向上させようと熱心に努力している。もちろんそれ以外の学年に対して期待していないわけではないが、やはり中 1 は何かこう素晴らしいものを持っていると思う。

で、何がいいたいかというと活きのいい新入部員がほしいという事だ。もしこの部に興味を持ったならば中学生が密集しているほうの校舎の 4+0.1 階くらいにある小部屋を訪れてほしい。パソコン部は歓迎してくれるだろう。少なくとも自分は大歓迎だ。